H. ZIMMERMANN

M. ELIE

PURPOSE.

Interconnecting Computer Networks of different organizations (nations, manufacturers, administrations, companies, etc...) raises the need for a well defined and largely accepted Host-Host protocol. Several discussions are being conducted within various national or international bodies, with the experience of several computer networks.

The present paper is an attempt to define such a protocol that matches most of the requirements encountered in the related discussions.

RECEIVED AT NIC JANUARY 16, 1974.

.../...

MULTIPLEXING - DEMULTIPLEXING.

Transport stations will exchange data units (DU), each DU concerning the flow identifier. The most recommendable option seems to be placing one and only one DU per packet. If strongly needed, multiplexing DUs within packets might be envisaged. Therefore, we shall use DUs in the following, where packet text should be more appropriate if no multiplexing is envisaged in the transport protocol. The size of a DU is anyway limited by the maximum size of the packet.

Transport stations also need to exchange information not related to flows, e.g. TS commands such as RESET, ECHO, etc... Therefore each DU will get an op. code :

- TS ... Op codes for TS commands
- FL ... Op codes for flows

The multiplexing-demultiplexing subfunction consists of sharing DUs between flows as well as TS commands.

Flow identifiers being permanent names (no dynamic allocation) in the transport function, the multiplexing-demultiplexing sub-function is a permanent one and can be used without any initial synchronization between transport stations.

FRAGMENTATION - REASSEMBLY.

- If the maximum size of the letter is bigger than what can be put into one single Data Unit, the origin transport station may need to fragment the letter into several Data Units and the destination transport station reassemble the letter from those Data Units (remember we don't assume the communication function to guarantee sequencing of packets).

- Therefore each letter is decomposed into several (possible one) fragments, each fragment going in one Data Unit. Fragments are numbered and a special flag "end of letter" (EOL) indicates the last fragment of the letter. The corresponding parameters in the Data Unit are the following

  - FR ## Fragment number within the letter
  - EOL Current/Last fragment of the letter
  - FR TEXT Fragment of text of the letter

  When every fragment of a letter has been received, the transport station can reassemble them and deliver the entire letter to the destination transport user.

  To make reassembly easier, the protocol imposes fragments within one letter to be all the same size (except the last fragment of the letter). If the letter is repeated (see error control on letters) the same fragment size must be used by the origin transport station.

.../...

Of course, the size of the fragment is limited by the maximum size of the data unit.

- The destination station should not wait indefinitely with a partially reassembled letter Therefore the reassembly process will be protected by a reassembly time-out. The time-out will be set when receiving the first (in time, but may be it is not ≠≠ 1) fragment (i.e. when starting reassembly) and will be reset when reassembly is successfull (i.e. all fragments received). If the time out occurs, reassembly will be aborted and the letter declared erroneous and dropped.

The problem arises not to mix fragments of different letters of the same flow (fragments of different flows cannot be mixed because the Data Unit contains the Flow Identifier).

Therefore, each letter gets a reference from the origin transport station. That reference must be "unique enough" to ensure a meaningless (e.g. 10^{-10})probability of mixing fragments of different letters. The choice of the reference may vary according to the type of service offered :

. if error control of letters is offered on that flow, the same reference will be used by both (fragmentation - reassembly and error control) sub-functions.

. else, one single cyclic reference generator may be used for all flows in one Transport Station.

The data unit containing a fragment will then contain another parameter :
. LT REF Letter "unique" reference

Duplicated fragments will be detected and dropped by the destination transport station, provided the letter is still under reassembly. If not, a late fragment may be considered as one of a new letter. discovered erroneous when the reassembly time out will occur.

The fragmentation reassembly sub-function is a permanent one and can be used without any initial synchronization between transport stations.

ERROR CHECKING ON BITS.

If the error rate on bits in the communication function is not low enough for some application, the transport user may require an additionnal end-to-end control on bits for one flow.

A checksum will then be added to the data units concerning that flow. Erroneous data units will be dropped *, thus transforming an error on bits in a loss of packet. If error control on letters is performed on that flow, that loss will be recovered by the transport function.

ERROR CONTROL ON LETTERS.

When in use on one flow, that subfunction is responsible to detect errors on letters and perform recovery.

---

* If an error occurs in one DU, it may be located in the opcode, in the address field or in the text as well. Therefore, the best solution is to drop it and not to start any processing with it.

.../...

The origin transport station uses a cyclic reference generator dedicated to that flow. Thus, the next letter in the flow gets the next reference.

Every received letter must be acknowledged either positively or negatively by the destination transport station.

. The position ACK of a letter means that all the letters with preceding references have been received without error.

. The negative ACK of a letter means an error has been detected on that letter and the destination transport station has dropped it.

The origin transport station expects every letter to be acknowledged within a maximum time. If no ACK is received within that time, the letter will be supposedly lost, and sent again. That process will be repeated "n" times. If not successful, the "error control on letters" subfunction will quit, warning the user with an unrecoverable error.

Repeating letters may create duplicated letters that will be detected, dropped and positively acknowledged * by the destination transport station.

The origin transport station is responsible for the uniqueness of the references given to the letters it sends. That means a reference must not be reused if a letter with the same reference (in the previous cycle) may still be in transit (i.e. the cyclic generator has a limit in cycling speed). The destination transport station will usually be prepared to receive references within a window smaller than the whole cycle, corresponding typically to the usual delay dispersion of the communication function or to its own possibilities if smaller. This contributes to decrease the probability for a duplicate of the previous cycle to be considered as a new letter.

Practically, the reference field size must be choosen according to the characteristics of the communication function and the maximum possible flow between transport users.

Error control on letters requires the initial synchronization of senders' cyclic reference generators with receivers' windows. A generator and a window are needed at both ends of the flow (full-duplex).

FLOW CONTROL.

Flow control may be added to error control on letters. The sender will ask for credits allocated by the receiver. A credit represents the permission to send one letter. The maximum size of the letter must have been decided in the initial synchronization (see below).

Demands and allocations of credits will refer to a letter reference in the flow. The corresponding parameters are :

. LT REF Letter reference
. CRD ## Number of credits demanded or allocated

A demand for credits means :

"May I send the letters with references going from LT REF to LT REF + CRD ## ? "

An allocation of credits means :

"You may send the letters with references going from LT REF to LT REF + CRD ## "

.../...

* The ACK may have been lost.

Termination may also be decided in 1 or 2 because of user request for termination, time out or disagreement on parameters.

A transport station that is implemented to offer only the permanent services should at least be able to answer to a synchronization request by a termination.

PARAMETERS NEGOCIATION.

A successfull initial synchronization requires an agreement on the parameters of the session, namely :

- ERROR CTRL ON LT : Yes or no
- FLOW CTRL ON LT : Yes or no, and if yes :
- LT MAX SIZE in each direction : Each TS will indicate the [illegible]uired maximum size for letters it will send(ORIG LT MAX SIZE). I[illegible] this is acceptable for the other party, the agreement is reached for flow control. Later on, when allocating a credit, the receiver must be prepared to receive a letter of the maximum size.

TRANSPORT STATION DATA UNITS.

Transport stations need to exchange informations not related to one flow. The corresponding data units an functions that might be envisaged to start with are :

TS-ECHO/TEXT : the text must be sent back to the origin transport station by a TS-REPLY-TEXT data unit.

TS-RESET : used to reset all mechanisms, tables, etc... concerning their relations in both origin and destination transport station. The RESET succeeds by the rendez-vous of two such DUs, one sent and one received. That synchronization must of course be protected by a time-out.

TS-ERROR/ERROR CODE/HEADER OF THE ERRONEOUS DATA UNIT : used during debugging phases, to indicate to a distant TS that a DU it sent had an error.

DATA UNITS CONTENTS.

The various data units with the corresponding parameters (may be optionnal are listed below) :

* TS ECHO/TEXT

  Use : request the DEST-TS to send back the TEXT in a TS-REPLY data unit.

  TEXT : bit string to be sent back.

* TS REPLY/TEXT

  Use : Answer to a TS-ECHO data unit

  TEXT : copy of the text received in the TS-ECHO

* TS RESET

  Use : to request a reset from the DEST-TS or to acknowledge a received TS-RESET. Tables, mechanisms, etc... concerning the other party must be reinitialized when one TS-RESET has been sent and one received (in any order).

.../...

FR-TEXT : Text of fragment

DU CHKSUM : Checksum associated with the DU

The control information in a FL TRAN data unit must be interpreted by the destination TS, even if the FR-TEXT must be dropped (no buffer).

* FL EVNT/ $\left\{ \begin{array}{l} \text{FL-ID} \\ \text{DEST PORT ≠≠} \end{array} \right\}$ / EVNT TEXT

Use : To send events on one flow.

FL-ID : See FL-TRAN

DEST-PORT ≠≠ : See FL-TRAN

EVNT TEXT : Text of the event.

PARAMETERS FORMAT.

OP-CODE :	4 bits
COMPL : complement to the op-code indicating options used	12 bits
TU ≠≠	16 bits
TU PORT ≠≠	8 bits
TS PORT ≠≠	8 bits
LT REF	8 bits
FR ≠≠	7 bits
EOL	1 bit
FR TEXT	n x 8 bits
CRD ≠≠	4 bits
EVNT TEXT	32 bits
CHKSUM	16 bits

DATA UNIT FORMAT.

The COMPL parameter could be used to indicate which fields are present in the DU, thus allowing to have variable format adapted to the parameters used. We might also consider more op-codes with fixed formats.

TRANSPORT USER INTERFACE.

That interface should allow the user to send and receive letters and events within his flows and request special services when needed. The primitive interactions between transport station and transport user can be summarized as follows :

.../...

	TU to TS	TS to TU
Sending a letter	FL-ID LT-TEXT	
Receiving a letter		FL-ID LT-TEXT
Sending an event	FL-ID EVNT-TEXT	
Receiving an event		FL-ID EVNT-TEXT
Enabling special services	User Request or Acceptation or Refusal	Distant Request or Acceptation or Refusal
Disabling special services	User Request	Distant Request

Though not listed here, error conditions might of course come back from TS to TU.

Each TS must know its TUs, some of which may be active, some others inactive. That knowledge may be passed to the TS during interactions with TUs through their interface or may be in other cases part of the TS code (static knowledge).

ACKNOWLEDGMENTS.

This set of proposals is certainly not the final say on the subject. However, it attempts to evolve a synthesis of various earlier studies. Full considerations have been given to software manufacturer's problems, since host-host protocols will presumably be integrated in the core of future operating systems. In particular, several improvements were introduced after stimulating discussions with E. ANDRE, B. JACQUIOT and J.F. LASSALLE (CII). Various concepts and mechanisms took shape in the course of discussions within IFIP-TC6-1 (INWG), and have already been formulated by V. CERF and B. KAHN (ARPANET). Substantial inspiration is due to D. DAVIES (NPL), whose pioneering work is permeating most present day network issues. At last, L. POUZIN's (CYCLADES) keen and sensible views on networking deeply influenced the fundamental options of that design.

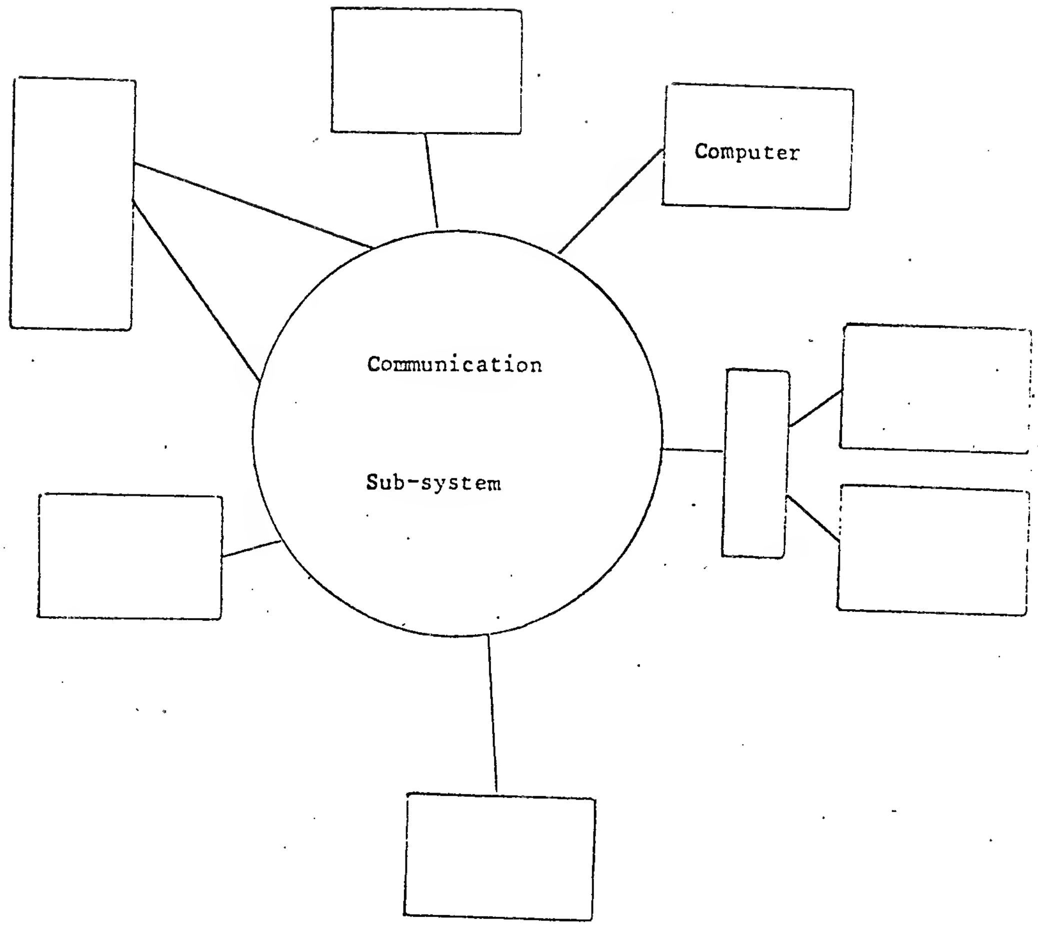

FIGURE 1

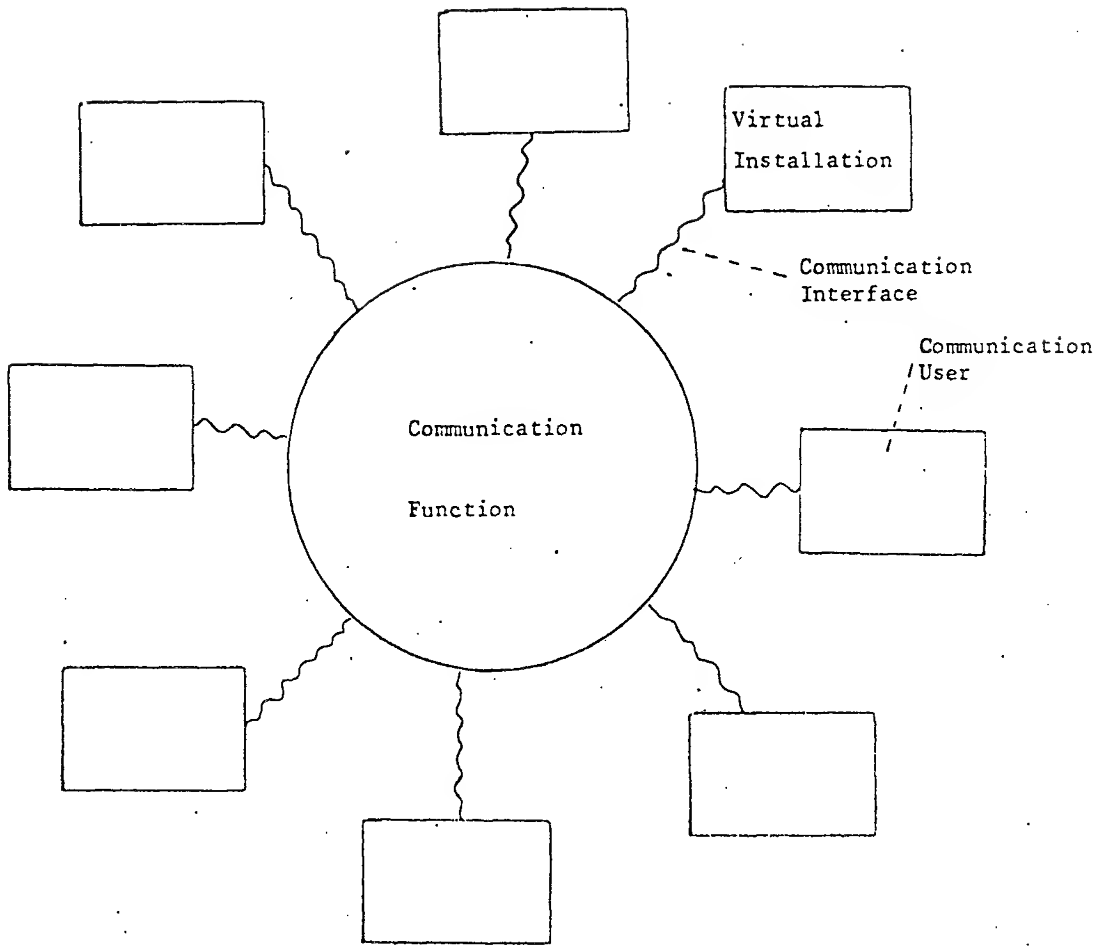
Virtual
Installation
Communication
Interface
Communication
User
Communication
Function

FIGURE 2

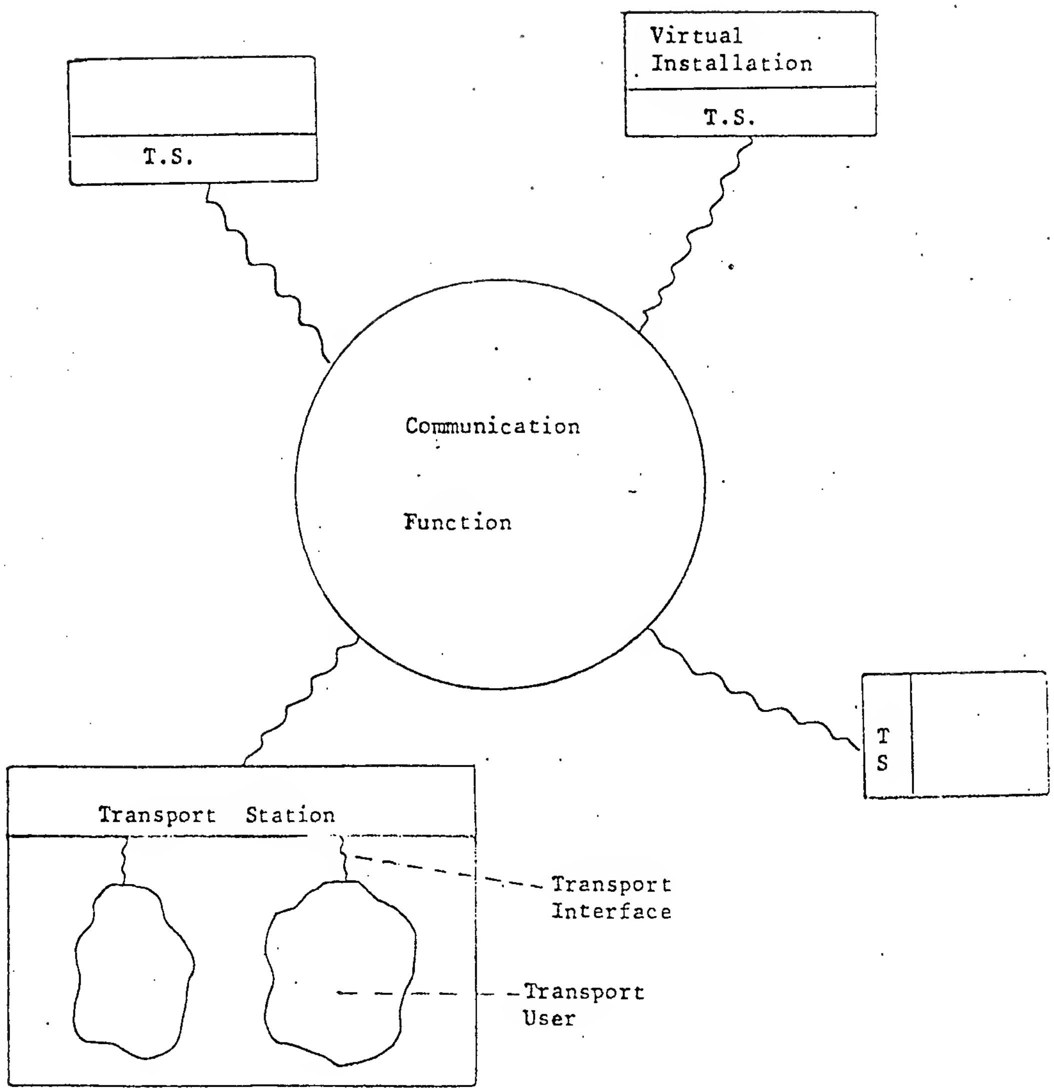
Virtual
Installation
T.S.
T.S.
Communication
Function
T
S
Transport Station
Transport
Interface
Transport
User

FIGURE 3

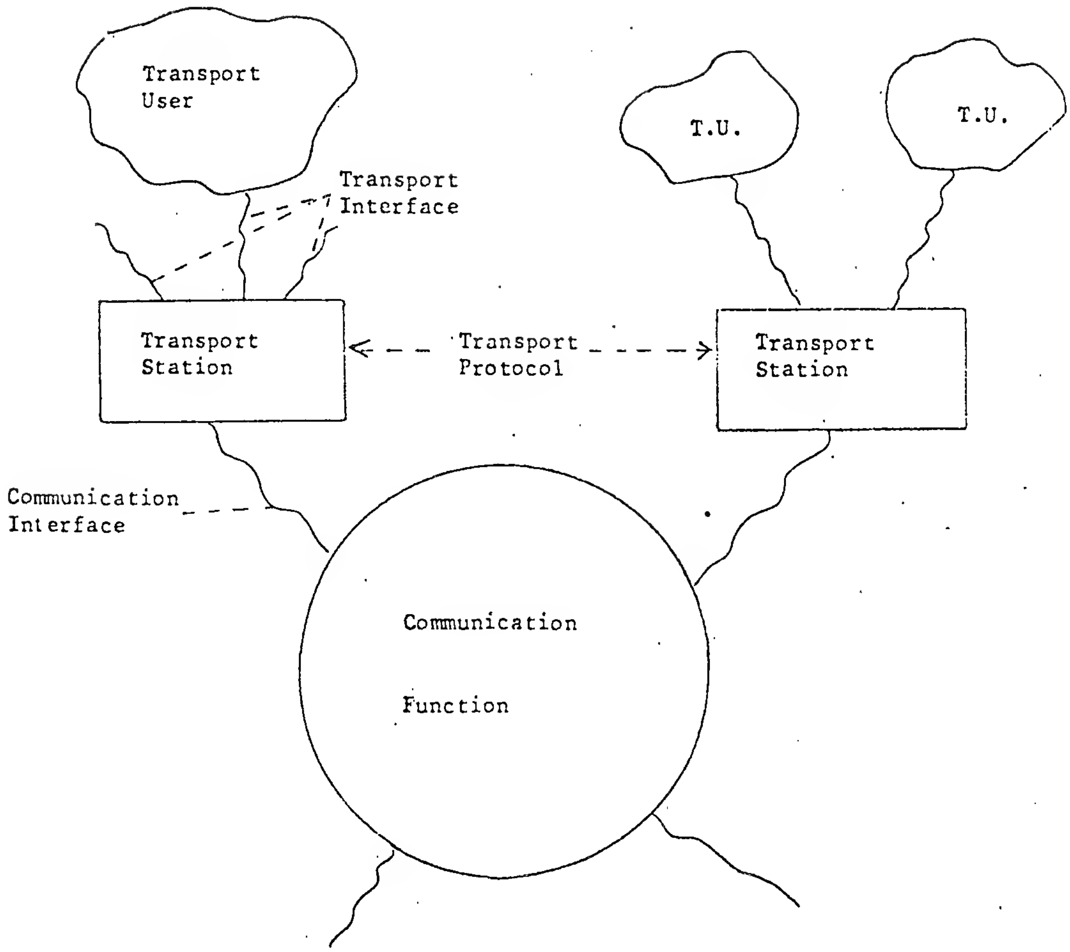
Transport User
T.U.
T.U.
Transport Interface
Transport Station
Transport Protocol
Transport Station
Communication Interface
Communication
Function

FIGURE 5

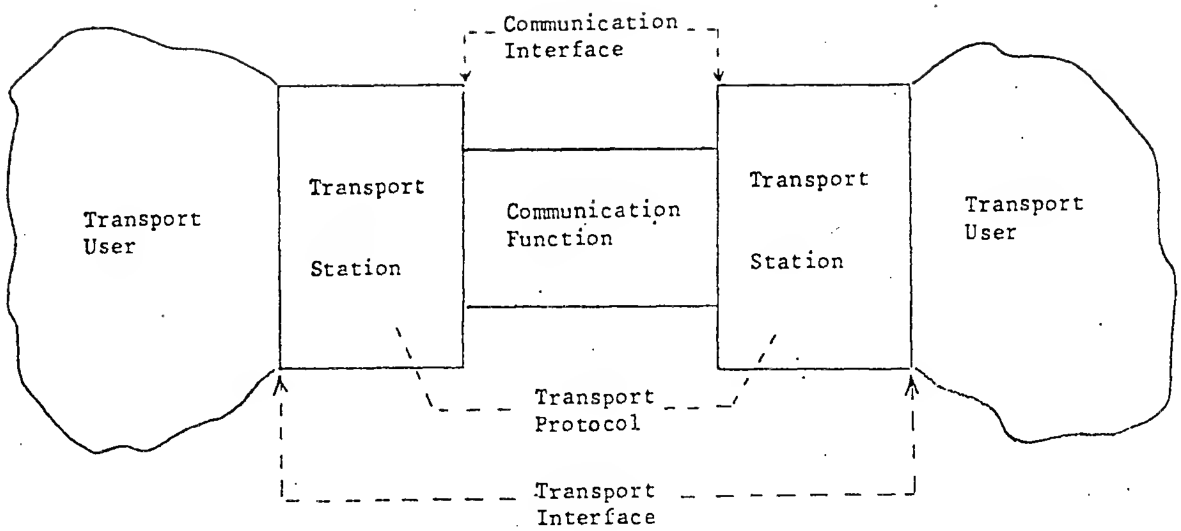
Communication
Interface
Transport
User
Transport
Station
Communication
Function
Transport
Station
Transport
User
Transport
Protocol
Transport
Interface

FIGURE 6

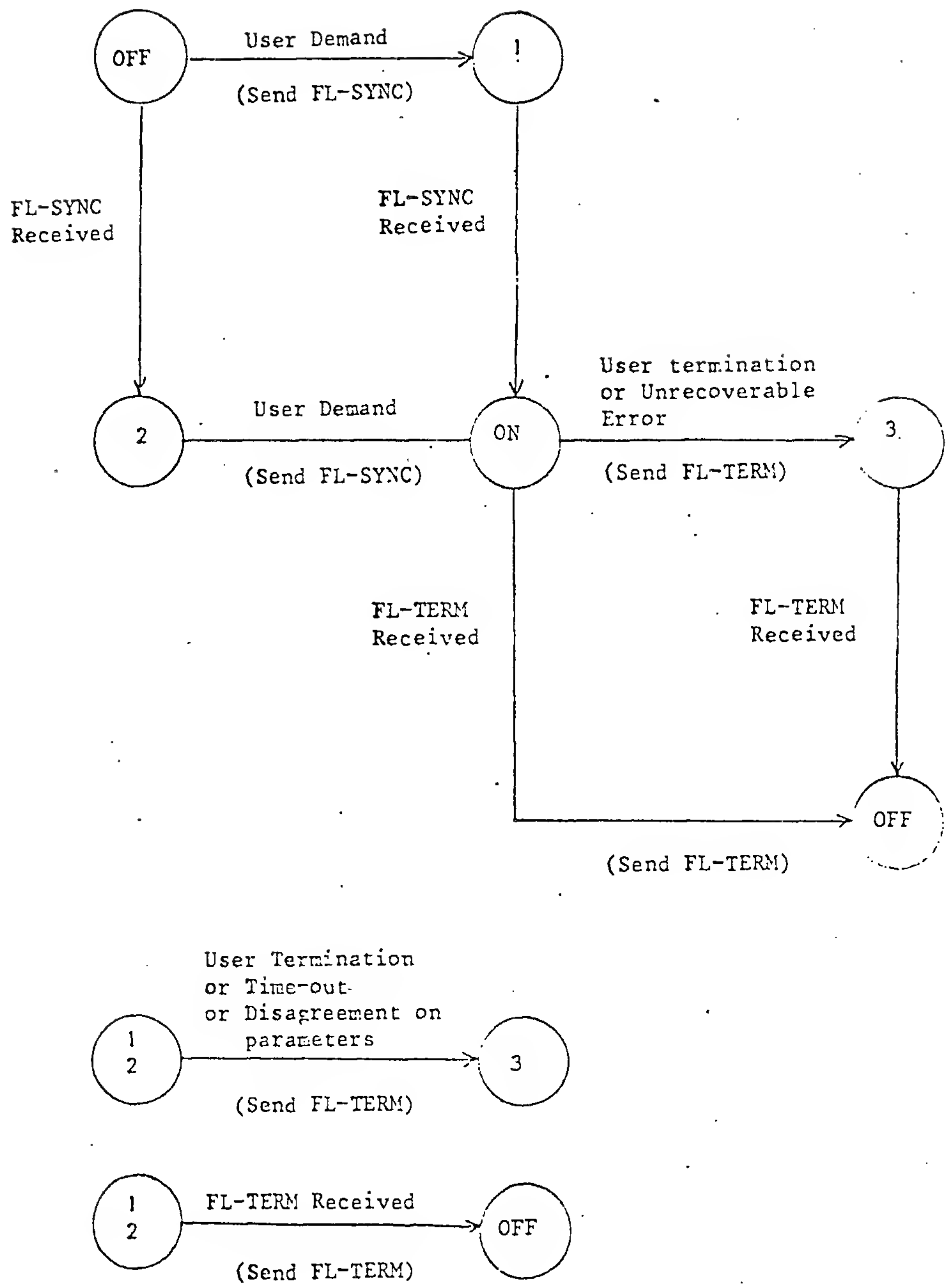
OFF
User Demand
(Send FL-SYNC)
1
FL-SYNC Received
FL-SYNC Received
2
User Demand
(Send FL-SYNC)
ON
User termination or Unrecoverable Error
(Send FL-TERM)
3
FL-TERM Received
FL-TERM Received
OFF
(Send FL-TERM)
1 2
User Termination or Time-out or Disagreement on parameters
(Send FL-TERM)
3
1 2
FL-TERM Received
(Send FL-TERM)
OFF

FIGURE 7